实验4 寄存器堆与运算器设计实验

一、实验目的

- 学习寄存器堆的结构和数据传送原理，掌握三端口寄存器堆的设计方法；
- 掌握运算器的结构与工作原理，能将寄存器堆与暂存器及ALU进行正确连接，构成运算器。

二、实验内容与原理

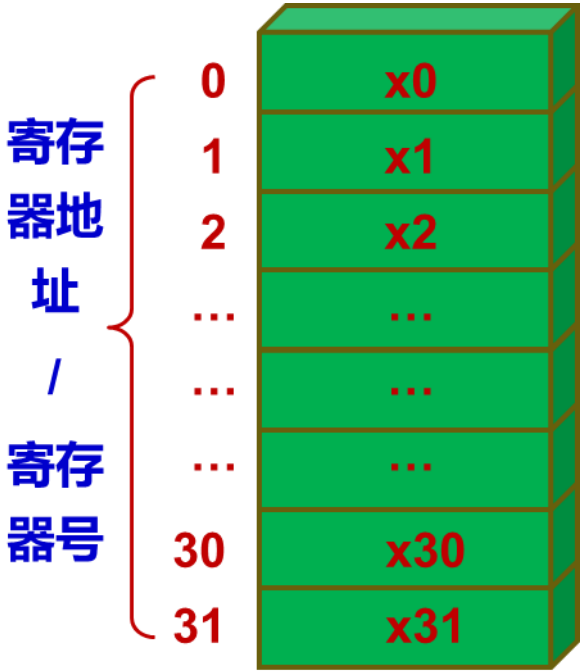
- **实验内容：**
  - 设计一个RV32I指令集架构的寄存器堆（32×32位）；
  - 将其与前述实验中的暂存器A、B、F和ALU进行连接，构成一个完整的运算器。

1、寄存器堆

2、运算器

1、寄存器堆

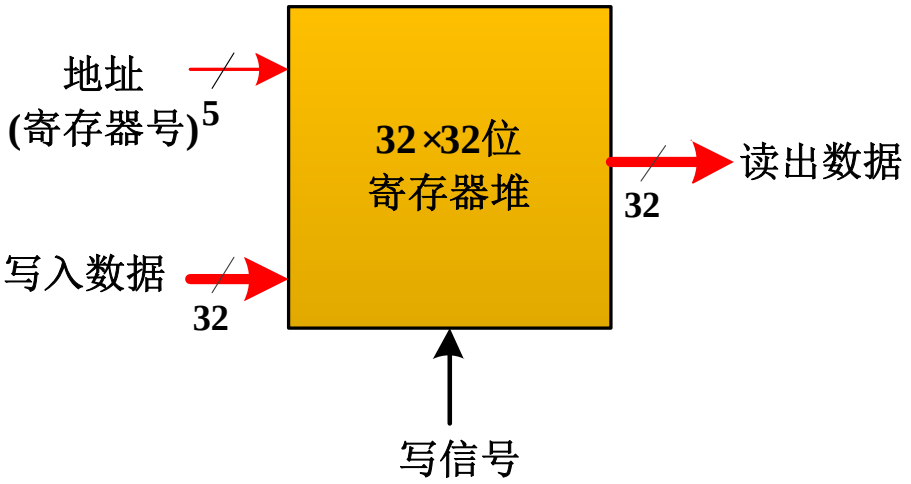
- **寄存器堆：寄存器集合**
  - **寄存器地址：寄存器号，对其中的寄存器进行统一编码**
  - **读写访问：通过指定寄存器号来进行读写访问，譬如R5，或者x5**
- **RISC-V的寄存器堆：32×32位**
  - 32（=2^5）个寄存器，每个寄存器32位
  - 寄存器编号（地址）：5位；
  - 寄存器内容（读写数据）：32位



1、寄存器堆

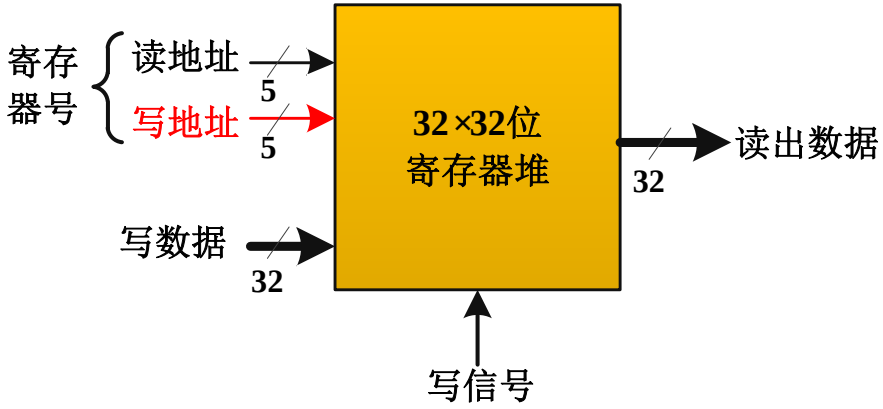
- （1）怎么定义端口?

- ①单端口寄存器堆：
  - 一个5位地址
  - 操作：读一个数据或者写一个数据

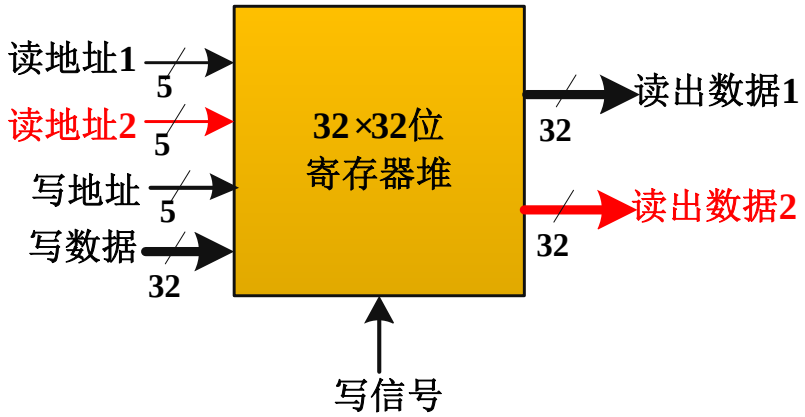


1、寄存器堆

- 双端口寄存器堆：
  - 两个地址：读地址和写地址
  - 操作：可同时进行读一个数据和写一个数据

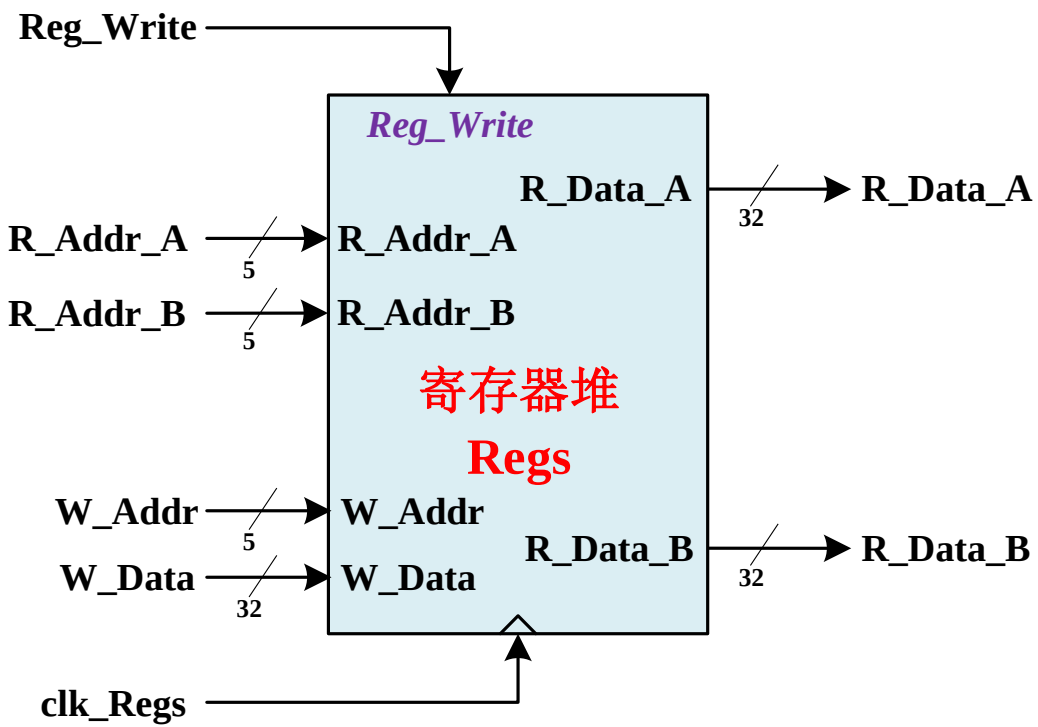


- 三端口寄存器堆：
  - 三个地址：两个读地址和一个写地址
  - 操作：可同时读两个数据、并写一个数据

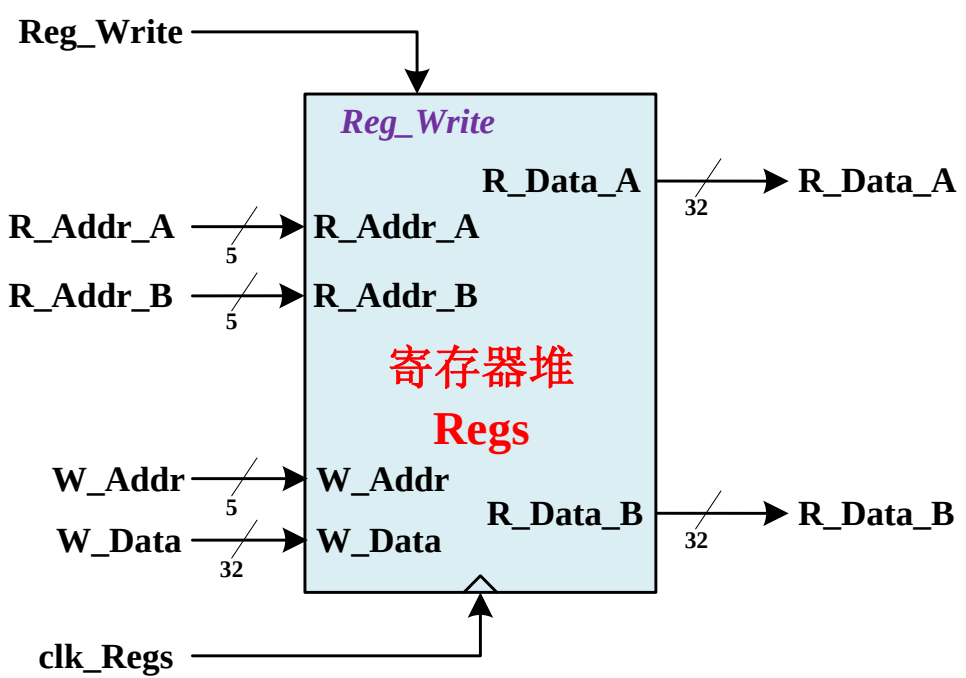


1、寄存器堆

- （2）本实验目标：三端口寄存器堆
  - 双端口读：2个读端口
  - 单端口写：1个写端口
  - 可并行操作
  - RISC-V的x0只能读全零



1、寄存器堆

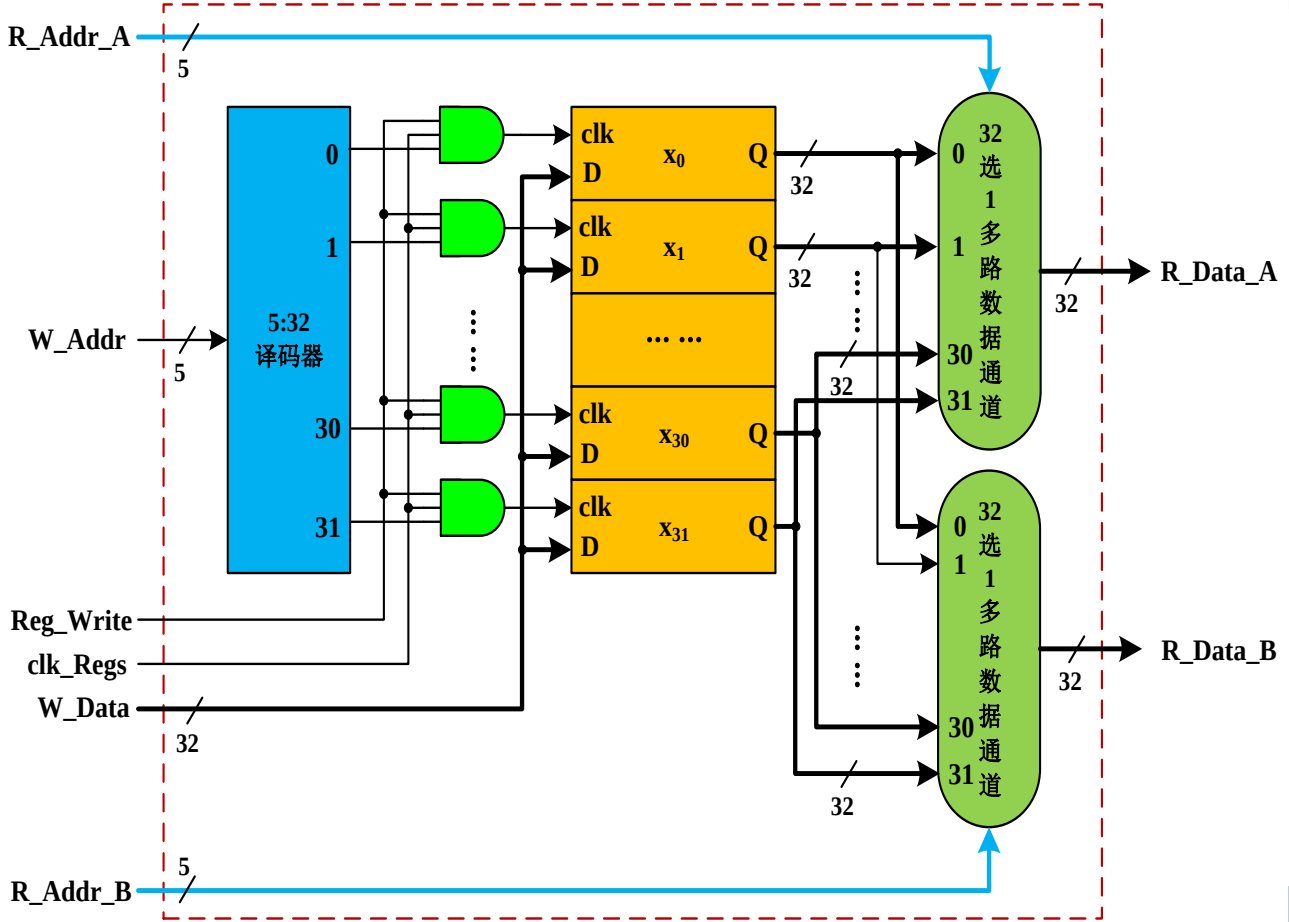


功能表

端口	时钟clk_Regs	地址输入	数据输入	数据输出	Reg_Write	操作
A端口	x	R_Addr_A	——	R_Data_A	x	读A口数据
B端口	x	R_Addr_B	——	R_Data_B	x	读B口数据
W端口	↑	W_Addr	W_Data	——	1	写入数据

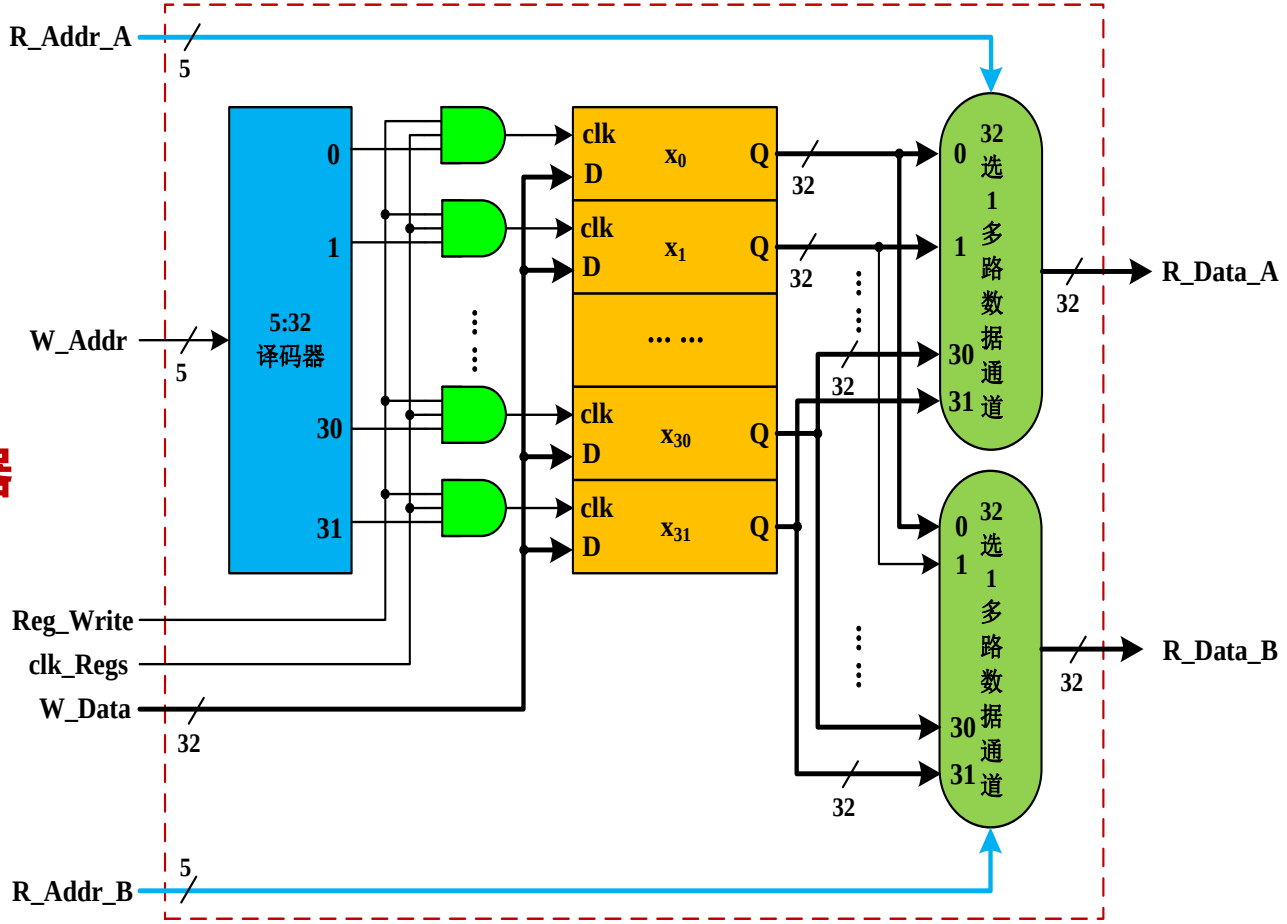
1、寄存器堆

- （3）内部结构：
- ①寄存器阵列
  - 每个寄存器由32个边沿D触发器构成
  - 32个寄存器各自独立



1、寄存器堆

- （3）内部结构：
  - ①寄存器阵列
  - ②写地址译码器
  - ③写入脉冲产生电路
  - ④读出数据多路选择器
    - 32位数据
    - 2个32选1选择器



1、寄存器堆

- （4） Verilog 实现
  - ①寄存器阵列：reg类型信号的数组

每个元素32位　32个元素

```
reg [31:0] REG_Files[0:31];
```

  - 寄存器地址译码：地址作为数组下标
  - ②读操作：组合逻辑

两个并行读操作

```
assign R_Data_A = REG_Files[R_Addr_A];
assign R_Data_B = REG_Files[R_Addr_B];
```

1、寄存器堆

（4） Verilog 实现：

- ③写操作：时序逻辑
  - $\overline{rst}$：用于初始化寄存器（全部清零）;异步信号；可采用CPU的复位信号
  - clk_Regs：时钟边沿信号，用于写入寄存器
  - Reg_Write：电平信号，用于指示写操作

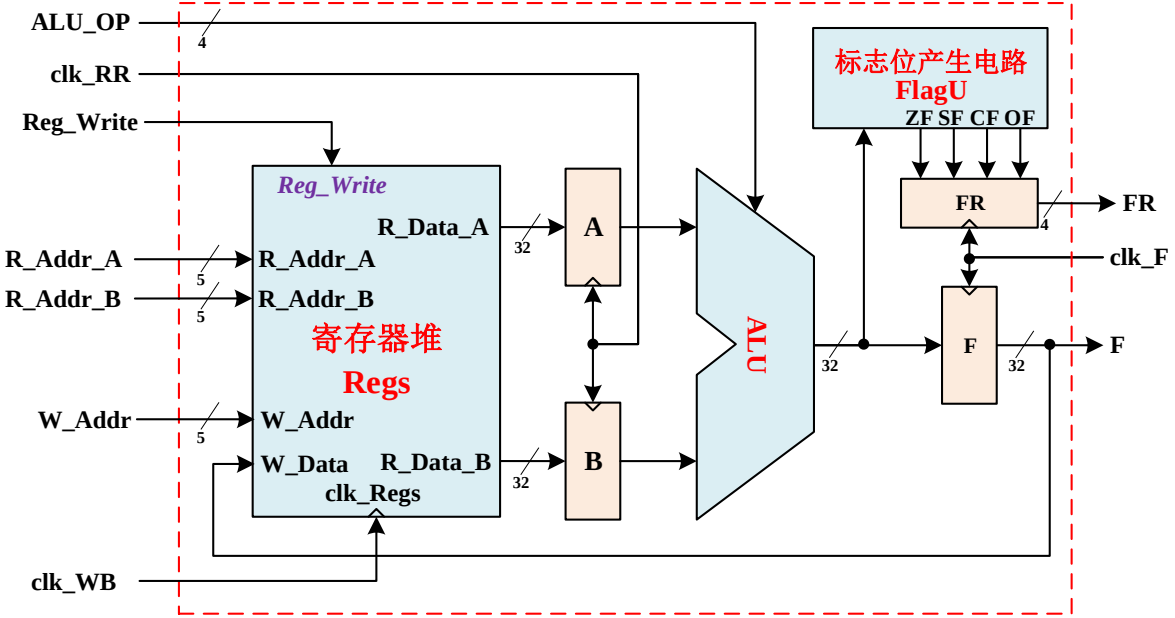
```
always @(posedge clk_Regs or negedge rst_n)
begin
   if(!rst_n) //低电平有效，=0则初始化
      ……;     //初始化32个寄存器
   else
   begin      //clk上跳沿
     if (Reg_Write)  //电平信号
       REG_Files[W_Addr] <= W_Data;
           //写入寄存器;
   end
end
```

2、运算器

- **运算器构成：**
  - ALU：核心部件，算术运算、逻辑运算；
  - 通用寄存器堆：存储运算数据、结果，程序员可见；
  - 暂存器：暂存运算数据和结果，程序员不可见；
  - 状态寄存器：记录运算结果的状态；
  - 连接部件：总线或者直接通路；

2、运算器

- **本实验的运算器：**



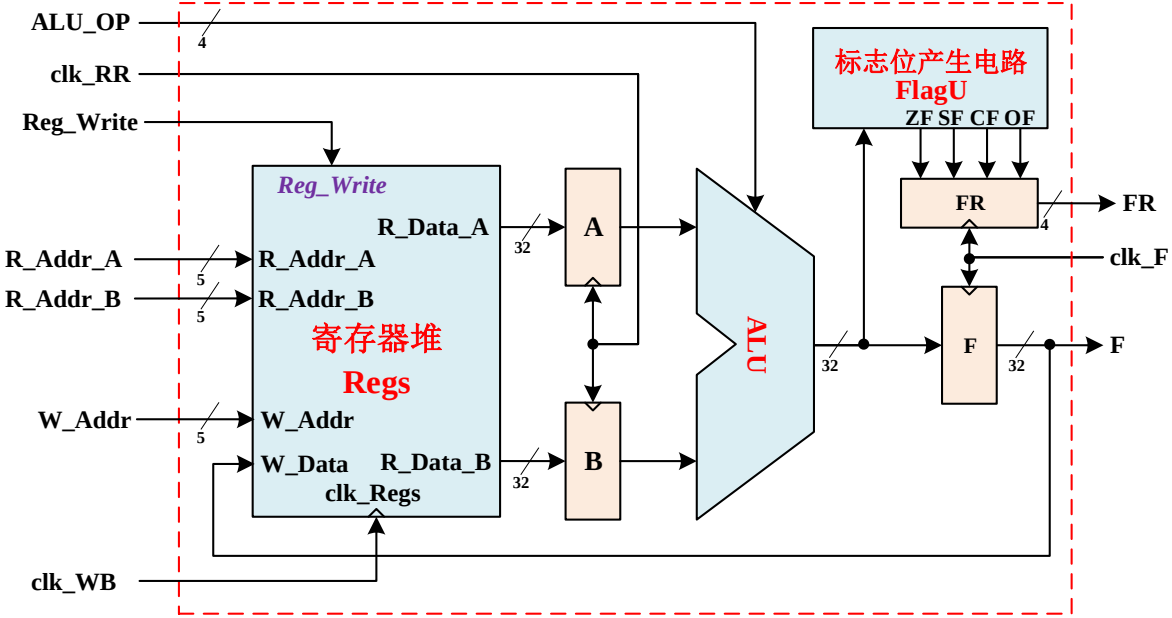
- **ALU：10种功能；**
- **通用寄存器堆：32×32位，三端口；**
- **暂存器：A、B、F，接寄存器堆的三端口；**
- **状态寄存器：FR，记录ZF、SF、CF、OF；**
- **连接部件：直接通路；**

2、运算器

- **本实验的运算器：**
  - **可完成什么运算？**

$$(x_i)\theta(x_j) \rightarrow x_k$$

  - x_i和x_j：源寄存器
  - x_k：目的寄存器



1st
① 给出寄存器堆A口地址R_Addr_A=i：R_Data_A=Regs[i]=（x_i）
给出寄存器堆B口地址R_Addr_B=j：R_Data_B= Regs[j]=（x_j）
② 提供clk_RR时钟上跳沿：R_Data_A→A，R_Data_A→B

2nd
③ 给出ALU_OP：指定ALU运算功能θ
④ 提供clk_F时钟上跳沿：$(x_i)\theta(x_j)$→F

3rd
⑤ 给出寄存器堆写入口地址W_Addr=k，且置Reg_Write=1
⑥ 提供寄存器堆的写回时钟clk_WB：F→x_k

送出或者产生数据，和把数据存储到目的部件是不能分开的

三、实验要求

1. 设计一个寄存器堆模块，要求符合RV32I的寄存器堆特征，仿真验证其功能。
2. 构造一个运算器模块，引用实验2的带暂存器的多功能ALU模块、引用上述寄存器堆模块，并将它们连接起来。
3. 针对使用的实验板卡，设计运算器模块的板级验证实验方案，编写顶层测试模块。

	信号名称	配置设备管脚
输入	R_Addr_A[4:0]	5个逻辑开关
	R_Addr_B[4:0]	5个逻辑开关
	W_Addr[4:0]	5个逻辑开关
	ALU_OP[3:0]	4个逻辑开关
	Reg_Write	1个逻辑开关
时钟	rst_n	按键
	clk_RR	按键
	clk_F	按键
	clk_WB	按键
输出	F[31:0]	8个数码管
	FR[3:0]	4个LED灯

三、实验要求

4. 选择寄存器号和运算功能，验证你的运算器是否能正常工作，将实验结果记录到表格中，要求寄存器的读写操作和ALU的运算功能都被有效测试。

A	B	ALU_OP	功能θ	F	FR（ZF SF CF OF）
x_i =	x_j =			x_k =	

三、实验要求

5. 撰写实验报告（不做要求），格式见附录，重点内容包括：

- 对仿真结果进行分析；
- 描述你设计的板级验证实验方案、模块结构与连接；
- 说明你的板级操作过程；
- 分析记录下来的板级实验结果、得到有效结论。
- 请力所能及回答或实践本实验的“思考与探索”部分。

四、实验步骤

1. 新建一个工程，新建一个寄存器堆模块，注意x0为零寄存器。
2. 编写激励代码，仿真验证寄存器模块功能；分析仿真结果，确保读写操作正确。
3. 将实验11.3的带暂存器的多功能ALU模块的*.v文件（包括子模块的*.v文件）拷贝到本工程的源代码目录下，并将它们加入工程。
4. 新建一个运算器模块，引用实验2的带暂存器的多功能ALU模块，及上述寄存器堆模块，各定义一个实例，将它们连接起来，如前述图所示。如果有必要，可进行仿真验证。

为方便进行有效的测试，可以修改寄存器模块的初始化操作，不再全部清零，而是初始化相关寄存器为典型的测试数据。

四、实验步骤

5. 设计运算器模块**板级验证**的实验方案，然后据此编写一个**顶层测试模块**。
6. 可以依据实际需要，对顶层测试模块进行仿真测试，或者直接进入管脚约束环节。
7. 新建**管脚约束**文件，依据引脚配置表的提示和设计的板级验证实验方案，进行相应的**引脚配置**。
8. 生成*.bit文件，下载到实验设备的FPGA芯片中。

四、实验步骤

9. **板级实验**：按照你所设计的实验方案，操作输入设备、观察输出设备，一般过程为：

   1）按rst_n键复位；

   2）拨动开关输入A口和B口地址，按下时钟键clk_RR，将读出的A口数据和B口数据分别打入暂存器A和B；

   3）拨动开关选择运算功能ALU_OP，按下时钟键clk_F，保存结果到F；

   4）拨动开关输入写端口的地址，拨Reg_Write开关=1，按下时钟键clk_WB，将暂存器F中的运算结果写入指定寄存器；

   5）拨动开关选择输出的数据，观察LED灯或者数码管，记录实验结果到表1格中，并分析实验结果是否正确。

- **注意**：在板级实验中，要保证能观察或者经分析后确认：**运算结果已经更新了目的寄存器的值。**

五、思考与探索（1-4中至少完成1道）

1. 在仿真测试和板级测试中，你是如何确认寄存器堆被写入成功的？请具体说明。
2. 按照引脚配置表，输入信号需要的开关总计20个，请分析板级实验的过程，说明哪些开关可以分时复用？为什么？按照这样的思路，请计算最少需要多少个开关？
3. 运算器框图中，暂存器A和B的打入脉冲是clk_RR，暂存器F和标志寄存器FR的打入脉冲是clk_F，目的寄存器的打入脉冲是clk_WB，请问它们能合并为一个吗？如果能，请给出计算的过程；如果不能，请说明理由。

五、思考与探索（1-4中至少完成1道）

4. 运算器框图只能实现寄存器间的运算$(x_i)\theta(x_j) \rightarrow x_k$，如果要想同时完成寄存器$x_i$内容与一个立即数$m$进行运算，结果送目的寄存器$x_k$，即：$(x_i)\ \theta\ m \rightarrow x_k$，你觉得应该如何改造运算器框图？说明你的设计方案，以及运算器工作的过程。
5. 谈谈你在实验中碰到了哪些问题？又是如何解决的？